

Remarks

The present response is to the Office Action mailed the above-referenced case on Dec. 3, 2003. Claims 19 and 20 are presented below in their last amended or for examination. Claim 19 are rejected under 35 U.S.C. 102(b) as being unpatentable over the reference of Chow et al. (U.S. 5,642,511), hereinafter Chow. Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over Chow in view of Cain et al. (U.S. 5,014,138), hereinafter Cain.

Applicant has carefully studied the prior art presented by the Examiner, and the Examiner's rejections and statements of the instant Office Action. In response, applicant herein provides argument that not all of applicant's claimed limitations are anticipated by the prior art. Applicant points out and argues the key limitations recited in applicant's claims, as supported by the description of the specification, which appear to be misunderstood by the Examiner in his rejections and statements.

Firstly, applicant wishes to object to the Examiner's use of applicant's specific claim language when stating in the remarks of the Office Action what the reference teaches, although when the specific portion of the reference cited by the Examiner is carefully studied by applicant, the disclosure teaches something completely different. This is a common and misleading practice in the Office and should be discouraged. A more responsible approach would be for the Examiner to quote the actual reference teaching, and then argue how the teaching reads on applicant's claimed limitations. The reference teaches what the reference teaches, and should be presented exactly as such. An argument may then be fairly made as to whether (or not) the teaching reads upon applicant's claimed limitations.

Applicant substantially amended independent claim 19 in the prior response to more clearly recite the programming tool and Interactive Process

Module (IPM), specifically reciting an object-oriented programming tool enabled for constructing an IPM adapted for interaction with modules of an operating system (OS) for accessing and providing data in a multimedia call center (MMCC). In the instant Office Action the Examiner, in response to applicant's previous amendments and arguments, has presented the new reference of Chow for disclosing all of the limitations of applicant's claimed as previously amended.

The Examiner has stated that, regarding claim 19, Chow discloses applicant's claimed object-oriented programming tool enabled for constructing Interactive Process Module (IPM) adapted for interaction with modules of an operating system for accessing and providing data, comprising all of the limitations of applicant's claim. Applicant respectfully disagrees. Applicant argues that Chow fails to explicitly disclose applicant's invention because Chow teaches an alternative invention solving an alternative problem than that which is solved by applicant's invention, and therefore the goal and outcome, or result of the invention of Chow is also different, as well as the method and apparatus for achieving it, as will be detailed further below.

The object-oriented programming tool of applicant's invention is provided as a generic programming module which represents and conducts a defined business process, and has the ability to obtain data from other modules, such as interaction object modules of an operating system, and utilizing that data in the execution of its intended goal. The module is editable in the sense that steps and sub-steps may be altered, added, deleted, and so on, along with names, allotted times, action parameters, etc. Due to the generic nature of the module, a programmer may select an existing module and edit it to save as a new module, which may represent and conduct a completely different defined business process.

The goal in building the IPM of applicant's invention is to provide an interfacing process application capable of executing and performing an entire

business process from start to end according to CINOS constraints, time constraints and enterprise rules, and which may be used as a functional model for the business process that it represents. Each completed IPM is callable by the operating system (OS) with which it is adapted to operate, when needed, to perform its programmed function. The IPM is innovative in part because a generic application or model may be used for building several differing automated processes, and because it breaks down a process into tightly controlled tasks and sub-tasks that are executed in concert through interface with other systems or modules of the OS.

Referring now to the reference of Chow, the Examiner has stated that Chow discloses an object-oriented programming tool for constructing an Interactive Process Module (IPM) comprising applicant's claimed graphical interface (col. 3, lines 19-46, Fig. 5), wherein a plurality of code sets operate in the IPM, each adapted to completion of a specific task in the overall process, and entry of tasks with parameters by a programmer sequentially builds a process comprising multiple tasks to be performed in a requisite order dictated by the prerequisite relationship, and when complete, the IPM represents and conducts the process.

Applicant respectfully points out to the Examiner that Chow fails to teach a programming tool for constructing an Interactive Process Module (IPM); rather, Chow teaches a programming interface between a software programmer and the computer, which enables the programmer, through visual items such as sliders, buttons, and so on, to structure, build and define an end-user application, in this case a graphical user interface (GUI) to be used by another software application. The invention of Chow, as described in the background section (col. 3, lines 19-29), is a programming tool which facilitates the writing of programs by the

creation, movement, and interconnection of icons, e.g., through a user interface or GUI, as opposed to through the conventional textual programming.

The graphical interface of Chow does not comprise an input facility adapted for defining a task, definition including task identifier, or task description comprising activities performable automatically by the operating system, as taught in applicant's invention and specifically recited in applicant's claim 19. Chow teaches enabling the programmer, through the GUI, to create and control the behavior of elements, such as sliders and buttons, of a GUI application, and associate the GUI elements to their parent functions.

The "visual builder" of Chow performs the function of target and proxy object "tree editor" to simplify the building of the trees over conventional textual software coding processes, and to simplify the process of attaching or associating nodes, or objects of the tree to their parent. A run-time target object hierarchy tree, and corresponding create/edit/build proxy object tree are disclosed, the target objects (Fig. 4, 70, 72 etc.) representing the final actual run-time objects which are desired to be build, such as the scroll bar, window, form, etc., objects of the GUI application to be created, and the proxy objects of the proxy tree (Fig. 5, 70A, 70B etc.) representing "stand-in" or "proxy" objects, each having a corresponding target object at run-time.

There is a one-to-one relationship between the associated target and proxy objects of the target and proxy object trees. The proxy objects know the events and operations, which are available to the associated target object so as to provide the programmer a selection of functions to attach. The target object, in this case could be a slider button of a GUI application being created by the programming tool of Chow, and the proxy object contains the information pertaining to the events and operation of the slider button. The target object tree is a hierarchical representation of the various nodes of the program to be constructed, and the

invention of Chow simply provides a graphical interface for manipulating and associating the target nodes, utilizing the associated proxy nodes of the proxy object tree. Chow does not teach the capability for constructing, editing or storing the Interactive Process Module of applicant's claims.

In applicant's invention task objects may be loaded into the IPM through the definable input interface, as required by the programmer in order to set up the main and sub-tasks inherent to the process. For example, the IPM may contain certain objects or modules adapted for accessing certain data from other modules in the OS, while others inserted in the IPM may be adapted for accessing certain other databases or resources, or for performing other certain functions related to the process. The IPM of applicant's claims is capable of executing and performing an entire business process from start to end according to CINOS constraints, time constraints and enterprise rules, and may be used as a functional model for the business process that it represents.

Chow fails to explicitly disclose applicant's object-oriented programming tool constructing an Interactive Process Module (IPM), and further, Chow is not capable of constructing the IPM of applicant's claims. The invention of Chow teaches building target and proxy object trees and editing various properties of proxy objects of the proxy object tree, each proxy object associated to a corresponding target object. The proxy object accesses information from only one source, that is, its associated target object.

Applicant fails to see how this teaching reads on applicant's object-oriented programming tool constructing an Interactive Process Module as claimed. The rather common practice of Examiners in rejecting claims because prior art teaches alternative inventions that might accomplish the same or similar purposes does not provide prima facie rejections, and should be discouraged. To

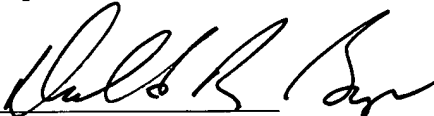
create a *prima facie* rejection, all of the actual elements of the claimed invention must be shown in the art. In this case the Examiner has clearly not done so.

Applicant therefore strongly believes that claim 19 in its present form is clearly and unarguably patentable over the reference of Chow. The Examiner has rejected depending claim 20 as unpatentable over Chow in view of Cain. In view of applicant's above arguments on behalf of claim 19, claim 20 is then patentable on its own merits, or at least as depended from a patentable claim.

It is therefore respectfully requested that this application be reconsidered, the claims be allowed, and that this case be passed quickly to issue. If there are any time extensions needed beyond any extension specifically requested with this amendment, such extension of time is hereby requested. If there are any fees due beyond any fees paid with this amendment, authorization is given to deduct such fees from deposit account 50-0534.

Respectfully Submitted,

Christopher Clemmett MacLeod Beck et al.

by 

Donald R. Boys
Reg. No. 35,074

Donald R. Boys
Central Coast Patent Agency
P.O. Box 187
Aromas, CA 95004
(831) 726-1457